

WebKit 内核源代码分析 (三)

红心地瓜(tomorrow.cyz@gmail.com)

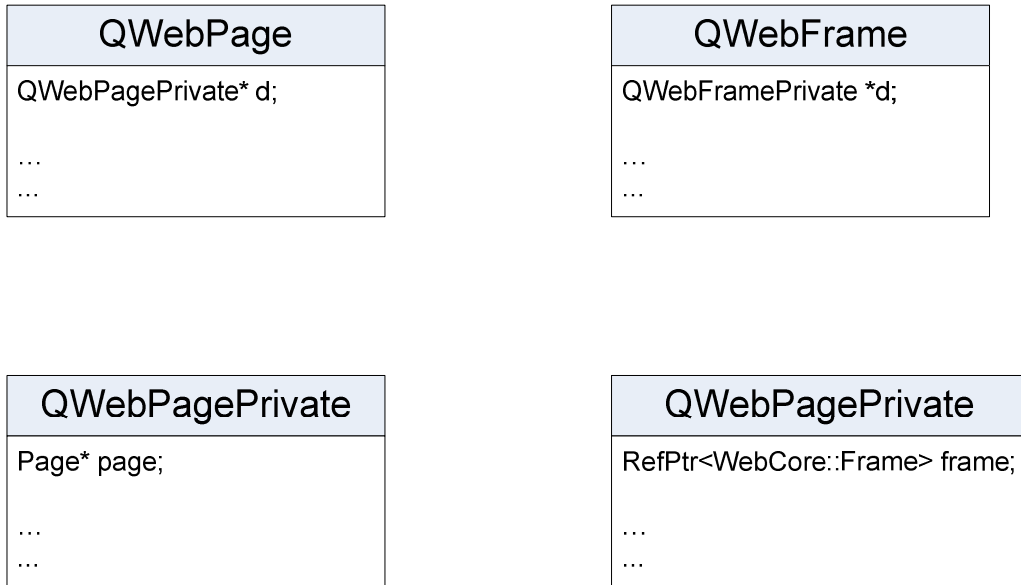
摘要：浏览器的请求一般是以页面请求为单位，当用户通过网址栏输入一个 url，浏览器就开始一个页面请求。而一个页面请求可能包含有一到多个页面子帧，以及图片、CSS 和插件等派生子资源。Page 类就是用来对应这样的页面请求。Page 类是 WebKit 中非常重要的一个类，它就像内核对外的一个聚合器。

关键词:WebKit 内核源代码,WebCore,Page,Frame，WebKit 架构

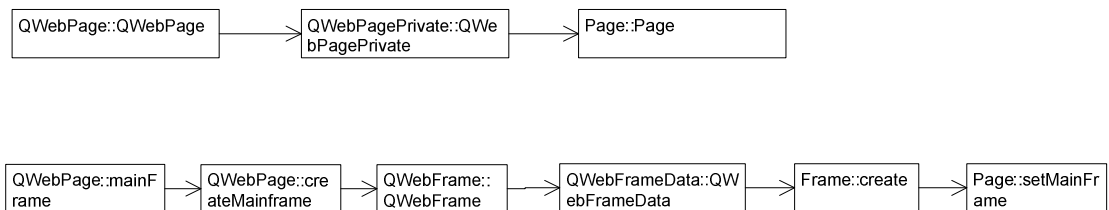
1. 概述

浏览器的请求一般是以页面请求为单位，当用户通过网址栏输入一个 url，浏览器就开始一个页面请求。而一个页面请求可能包含有一到多个页面子帧，以及图片、CSS 和插件等派生子资源。Page 类就是用来对应这样的页面请求。前进后退，导航，编辑，右键菜单，设置，Inspector 等这些用户参与的动作，大部分是同 Page 相关的。而标记语言解析、排版、加载则更多地同 Frame 相关。

我们通过几个图来看下 Qt 移植中 Page 类同应用之间的关系。

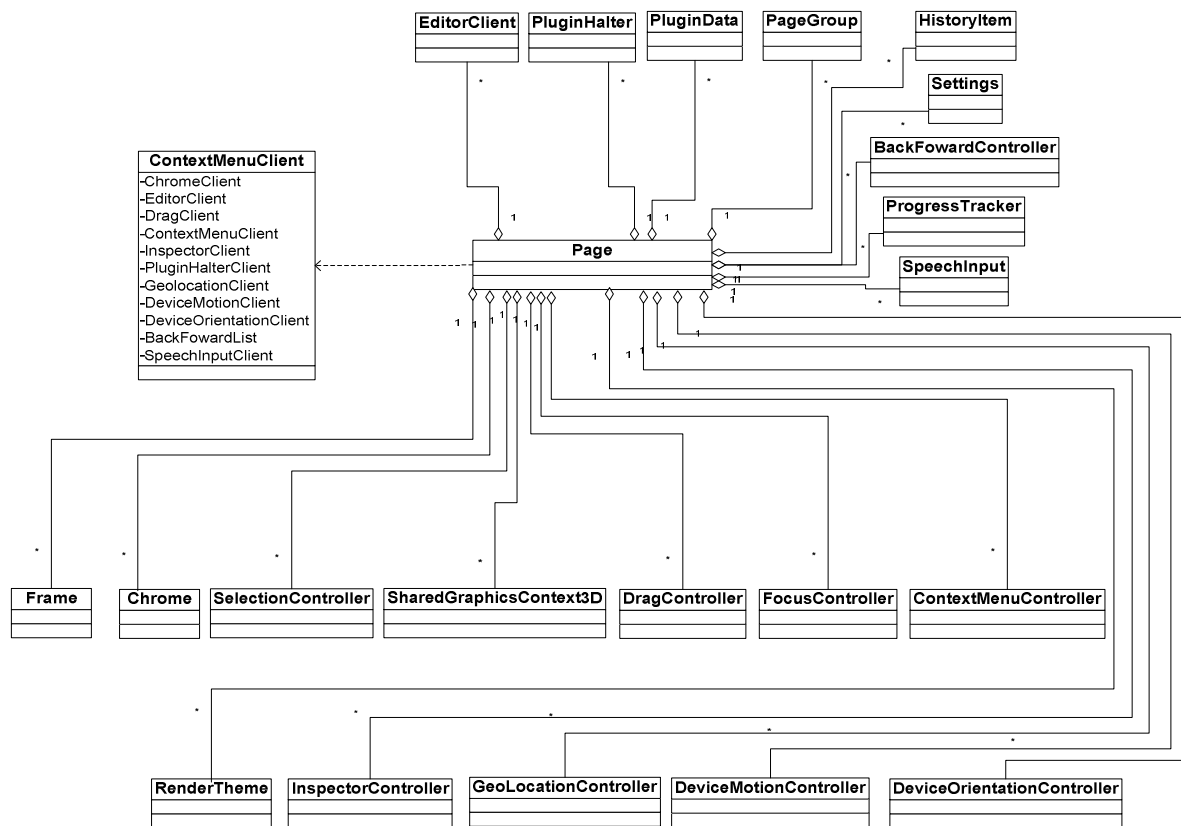


QWebPage 通过 QWebPagePrivate 维护 Page 类的指针，并在 QWebPagePrivate 的构造函数中实例化 Page 对象。QWebPage 类通过之后的 createMainFrame 调用实例化 QwebFrame，而在 QwebFrameData 的构造函数中，以 Page 指针为参数调用了 Frame::create 创建出 Frame 对象。



Page 类通过组合其它类的方式，实现了很多功能，Page 类本身并没有多少代码。

2. 类关系



2.1 PageGroup

PageGroup 并不是用来对 Page 进行管理的，而是设计用来将一些具有共同的属性或者设置的 Page 编成组的，以方便对这些属性进行管理。目前这样的属性包括 localStorage 的属性，IndexedDB，User Script，User StyleSheet 等。最常见的同 PageGroup 相关的操作是维护已访问链接（如 addVisitedLink 等接口）。根据地瓜的理解，假设 WebKit 内核之上架设多个应用（浏览器是一个应用），比较可能的是，一个应用独立一个 PageGroup。这里同多 tab 页没有关系，多 tab 页属于同一个 PageGroup。地瓜曾在 mailing group

上就这个问题咨询过，一位 RIM 的同学给我举了一个例子，比如一个基于 WebKit 的邮件程序，一方面他可能调用基于 webkit 的 browser 来显示网页，另外他本身也基于 webkit 来显示一些邮件，这两个之间的 setting 有很大可能不一样，他们就使用不同的 PageGroup。

PageGroup 中有这个 Group 已经安装并且使用的 User Script 和 User StyleSheet 的集合，一般在网页解析完毕后，这些 User Script 和 User StyleSheet 会插入到 Document 中。

PageGroup 中还维护了 Local Storage 和 Index DB 相关的设置，比如它们的 Path，上限等，通过 GroupSettings 类实现。

PageGroup 创建以后，每次创建一个新的 Page 对象，会通过 addPage 接口加入到这个 PageGroup 的 m_pages 中。

每次有导航行为发生的时候，会调用 addVisitedLink 来将 url 加入到已访问链接中。如果浏览器要跟踪已访问的接口，则在初始化的时候必须调用 PageGroup::setShouldTrackVisitedLinks，且参数为 true。此处 shouldTrackVisitedLinks 是一个静态的全局变量，也就是说，所有应用维护一样的行为（一个应用将其设置为 false 会影响到其它同样基于此核的应用）？

Page 类中维护了 PageGroup 的指针，并提供了 group 接口，这是个 lazy 接口，如果 m_group 不存在，会调用 InitGroup 来创建一个。对于 Page 类来说，如果没有设置 GroupName，则在初始化的时候会生成一个空 GroupName 的 PageGroup，由

m_singlePageGroup 维护，并把指针赋给 m_group，如果以非空的名字调用了 setGroupName，则会重新创建 PageGroup，此时这个 PageGroup 由 m_group 来维护。

2.2Setting

WebCore 中的设置相关的类，浏览器应用的不少配置、选项同该类相关，Qt 移植中，应用在创建 Page 对象后，会根据 Page::settings 来实例化 QwebSetting。

2.3Chrome

原生窗口接口类，参考地瓜写的“WebKit 中的 Chrome 和 ChromeClient”。

2.4其它

SelectionController:负责管理 Page 中的选取操作，绝大部分选取操作是基于 Frame 的，只在 Frame 的 Selection 为空的时候，对焦点游标的绘制工作才会使用到 Page 类的 SelectionController。

SharedGraphicsContext3D:共享 3D 图形上下文,为了优化 2D 显示而加入。在加速的 2D canvas 中，引入的 DrawingBuffer 的概念，SharedGraphicsContext3D 提供了 createDrawingBuffer 来创建 DrawingBuffer。

DragController:拖拽控制器。Chrome 的超级拖拽功能同这个有关？地

瓜会在以后对此进行求证。

FocusController:焦点控制器。考虑到焦点会在各个 frame 之间切换，所以由 Page 类维护焦点控制器最合适不过。

ContextMenuController:右键下拉菜单控制器。

InspectorController:Inspector 控制器，浏览器中的很多开发工具都同这个类相关。

GeolocationController:定位定位服务控制器。

DeviceMotionController:设备移动控制器

DeviceOrientationController:设备方向控制器

SpeechInputClient:语音输入 Client。

ProgressTracker:进度跟踪。

BackForwardController:前进后退操作控制。

Frame:一个 Page 由至少一个主帧和若干个其它子帧构成。

HistoryItem:历史记录。

PluginData:插件相关，未来可能同 PluginDatabase 类合并。主要是初始化 Plugin 的信息。

PluginHalter:用来控制 Plugin 的停止和重新开始。

RenderTheme:这个类提供了控件的渲染和绘制接口。Qt 移植中，RenderThemeQt 是 RenderTheme 接口的具体实现。

EditorClient:同编辑功能相关，比如拷贝、剪切、删除等操作。